Theoretical Computer Science 412 (2011) 6316–6326

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

A volume first maxima-finding algorithm☆

Xiangquan Gui[a,b,*], Xiaohong Hao[b], Yuanping Zhang[c], Xuerong Yong[d]

[a] College of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, PR China
[b] College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, PR China
[c] College of Computer and Educational Software, Guangzhou University, Guangzhou, 510006, PR China
[d] Department of Mathematical Sciences, University of Puerto Rico at Mayaguez, P.O. Box 9018, PR 00681, USA

ARTICLE INFO

Article history:
Received 7 February 2010
Received in revised form 5 May 2011
Accepted 5 August 2011
Communicated by S. Sen

Keywords:
Maxima
Skyline point
Computational geometry
Probabilistic analysis

ABSTRACT

The maxima-finding is a fundamental problem in computational geometry with many applications. In this paper, a volume first maxima-finding algorithm is proposed. It is proved that the expected running time of the algorithm is $N + o(N)$ when choosing points from CI distribution, which is a new theoretical result when the points belong to $d(>2)$ dimensional space. Experimental results and theoretical analysis indicate that the algorithm runs faster than the Move-To-Front maxima-finding algorithm.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

The maxima-finding, a fundamental problem in computational geometry, is closely related to the convex hull problem and arises in many applications such as Pareto-optimality in bargaining games or multi-criteria optimization, linear programming, unbounded knapsack problem and statistical decision theory [1]. Given a set S of N points in the d-dimensional Euclidean space E^d $(d \geq 2)$ with coordinates $x_1, x_2, \ldots, x_d$. A point p_1 dominates a point p_2 if and only if $x_i(p_2) \leq x_i(p_1)$ for $i = 1, 2, \ldots, d$. A point p in S is a maximal element of S if there does not exist any point q in S such that q dominates p and $q \neq p$ [2]. The set of all maximal elements of S is the set of maxima of S. The maxima-finding problem is to find such a set — the maxima of S.

Kung et al. showed that any algorithm that solves the maxima problem in two and three dimensions requires $\Omega(N \log N)$ time in the comparison-tree model. By the divide-and-conquer approach, they presented an algorithm to find all maxima for a set of N points in E^d, whose running time is $O(N \log^{d-2} N) + O(N \log N)$ [3]. When each point of the set has component independence (CI) distribution, Bentley et al. presented the FLET (Fast Linear Expected-Time) algorithm for computing the maxima set with a linear expected running time $O(N)$ [4]. A set of points has CI distribution if and only if all the d components of each point are chosen independently from continuous distributions. Although the FLET algorithm does have linear expected time, it may fail with probability $1/N$. So Bentley et al. presented another algorithm named MTF (Move-To-Front) and the empirical evidence of which implies that it runs faster than every other algorithm presented in the literature [4] and they conjectured that the MTF algorithm runs in $N + o(N)$ expected time. Subsequently, Golin proved that choosing

☆ The research was partially supported by Gansu Technology Support Project (No. 0804GKCA052), DIMACS and University of Puerto Rico at Mayaguez.
* Corresponding author at: College of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, PR China. Tel.: +86 13919056904.
E-mail addresses: xqgui@lut.cn (X. Gui), haoxh@lut.cn (X. Hao), ypzhang@lut.cn (Y. Zhang), xryong@math.uprm.edu (X. Yong).

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.08.013

N points from a 2-dimensional CI distribution, then with probability $1 - N^{-\Omega(\log N)}$, the MTF algorithm can find the maxima using only $N + O(N^{6/7} \log^4 N)$ expected time [5]. For the cases that the input points are chosen from higher dimensional spaces or from some other distributions, the conjecture is still open.

The maxima-finding problem has recently been studied in many areas. For example, as the skylines in data queries are just the set of minima in a given set of points, a number of maxima-finding algorithms have been proposed alone with studies of skyline queries, some of which are the BNL (Block Nested Loops) algorithm [6], the SFS (Sort Filter Skyline) algorithm [7], and the LESS (Linear Elimination Sort for Skyline) [8]. The SFS and LESS algorithms both can be considered as improved versions of the BNL algorithm. But they require the data to be topologically sorted in the beginning where just the sorting phase of the data set will cost $\Omega(N \log N)$ expected time.

In this paper we propose a new heuristic maxima-finding algorithm using the volume first (VF) heuristic instead of the Move-To-Front. Our experimental results show that it runs faster than the MTF algorithm. It can also be proven that the VF algorithm runs in $N + o(N)$ expected time when choosing points from CI distribution. More specifically we will prove that the expected running time of the VF algorithm is only $N + O(N^{2/3} \log^4 N)$ in the 2-dimensional space, which is better than the existing theoretical result of the MTF algorithm, and $N + O(N^{d/(d+1)} \log^{d+1} N)$ in $d(>2)$ dimensional space. This is a new theoretical result.

The rest of the paper is organized as follows. In Section 2 we introduce and discuss the VF algorithm and then in Section 3 we analyze the experiment results and show that the VF algorithm runs faster than the MTF algorithm. The expected running time of the algorithm is derived in Section 4. We address in the last section that for the non-CI distribution points the problem is still open.

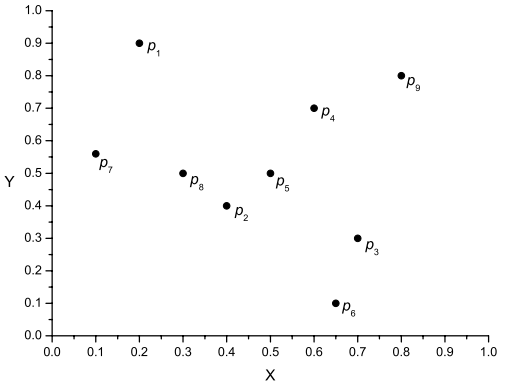
2. A volume first maxima-finding algorithm

Given a set S of N points in a d-dimensional space, many of the points in S are in general not the maximal points when N is large, while some particular points may quickly dominate many of those points. This insight has been used in the MTF algorithm. Inspired by the MTF algorithm, in this note we propose a new maxima-finding algorithm, which is called the VF algorithm, by using a volume first heuristic. The algorithm is easy to implement, very efficient for CI distributions and somewhat robust for points sets from other distributions.

The VF algorithm introduced is an on-line algorithm that maintains a temporary maxima sequence with a volume first heuristic. Its primary data structure is the sequence T of (indexes of) current maxima. The sequence T is originally empty, and at the conclusion of the algorithm it contains the maxima of S. The algorithm examines all input points in a random order. As the algorithm examines the input point Q, it compares Q with every point R in T. If R dominates Q, then the next of the input points is examined, as Q cannot be maximal. If Q dominates R, then R is removed from T. If Q is dominated by no R in T, then Q is inserted to a suitable position of T according to the *volume* of Q. In this paper we define the *volume* of a point to be the product of its coordinates' distribution functions in every dimension. Formally, given a point p with coordinates x_1, x_2 , …, x_d, and let $F_i(x_i) = Pr(X \leq x_i)$ be coordinates' distribution functions in every dimension, where $i = 1, 2, \ldots, d$ and the volume of p is equal to $\prod_{i=1}^{d} F_i(x_i)$. Distribution functions $F_i(x)$ normalize coordinates in every dimension to the [0, 1] region. This makes the product look like the volume of a cube that is determined by the point and the origin. Intuitively the bigger the volume of a point is, the more dominant *power* it owns. Therefore Q is inserted to T according to the descending order of the point volumes. In this way the maxima near the front of the sequence T tend to be more *powerful* to dominate the other points. They can quickly dominate most of input points which are not in the final output. If we do not know what the coordinate distribution function is, in some dimension of input points, we can simply set $F_i(x) = x$ for these dimensions. Although such a volume of a point is not good as the original one, it can make powerful points near the front of the sequence T. The pseudocode of the algorithm is described as follows:

VF Algorithm. the Volume First Maxima-Finding Algorithm.

```
TopMax:=1
Max[TopMax]=Point 1
for I:=2 to N do
    J:=1
    while (J ≤ TopMax) do
        if (point Max[J] dominates point I) then
            break //break while loop and exam next I//
        else if (point I dominates point Max[J]) then
            shift Max[J+1..TopMax] to Max[J..TopMax-1]
            TopMax:=TopMax-1
        else //point I and Max[J] are incomparable//
            J:=J+1
    if (J=TopMax+1) then
        K:=1
        while ( volume of Max[K]≥ volume of Point I and K≤ TopMax) do
            K:=K+1
```



i	New T	Point Volumes in T	Action
1	$[p_1]$	[0.18]	Start
2	$[p_1, p_2]$	[0.18,0.16]	Insertion of p_2
3	$[p_3, p_1, p_2]$	[0.21, 0.18, 0.16]	Insertion of p_3
4	$[p_4, p_3, p_1]$	[0.42, 0.21, 0.18]	Insertion of p_4; p_2 discarded
5	$[p_4, p_3, p_1]$	[0.42, 0.21, 0.18]	p_5 dominated by p_4
6	$[p_4, p_3, p_1]$	[0.42, 0.21, 0.18]	p_6 dominated by p_3
7	$[p_4, p_3, p_1]$	[0.42, 0.21, 0.18]	p_7 dominated by p_4
8	$[p_4, p_3, p_1]$	[0.42, 0.21, 0.18]	p_8 dominated by p_4
9	$[p_9, p_1]$	[0.64, 0.18]	Insertion of p_9; p_4, p_3 discarded

Fig. 1. A worked example in 2-dimensional space.

```
if (K=TopMax+1) then
    Max[K]:=Point I
else
    shift Max[K..TopMax] to Max[K+1..TopMax+1]
    Max[K]:=Point I
TopMax:=TopMax+1
```

In the pseudocode, T is maintained in the array Max[]. We present a full worked example with the point set chosen randomly in 2-dimensional area of coordinates $X \in [0, 1]$, $Y \in [0, 1]$. The points are shown in Fig. 1. The table illustrates how the contents of T is changed in the VF algorithm.

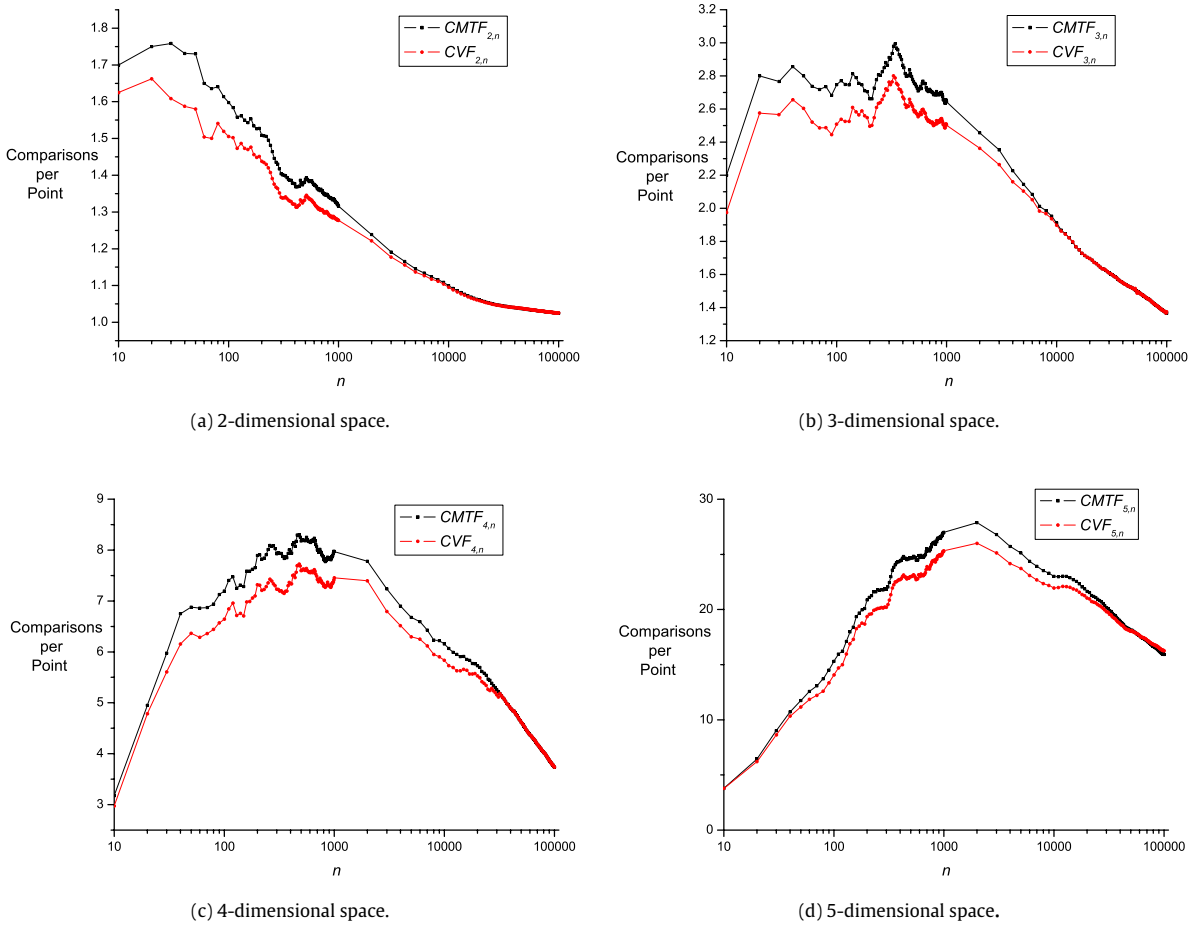
3. Experimental data

An interesting question is how fast the VF algorithm is. Obviously the algorithm runs in $O(N^2)$ time since T can contain at most $i-1$ points when p_i is examined. This is the same upper bound as the MTF algorithm has. However we believe that the VF algorithm has a better performance and moreover we will see that it is better than MTF from two aspects: experiment and theoretical analysis.

In experiment, we generate 10 input point sets, $S_1, S_2, \ldots, S_{10}$ with the same size $N = 100000$. Every coordinate of points is a double floating number in [0, 1] created randomly and the points are of dimension 5. Denote $S_i = \{p_{i1}, \ldots, p_{iN}\}$, $p_{ik} = (x_{ik1}, x_{ik2}, x_{ik3}, x_{ik4}, x_{ik5})$, where $i = 1, \ldots, 10$ and $k = 1, \ldots, N$, and let $S_{i,d,n}$ be a subset of S_i, such that $S_{i,d,n} = \{q_{i1}, \ldots, q_{in} : q_{ik} = (x_{ik1}, \ldots, x_{ikd}),\ k = 1, \ldots, n\}$, where $d = 2, \ldots, 5$ and $1 \le n \le N$ denote the point dimensionality of subsets and the point number of subsets, respectively. Then we run both VF and MTF algorithms on $S_{i,d,n}$, with $i = 1, \ldots, 10$, $d = 2, \ldots, 5$, $n = 10, \ldots, N$. The total numbers of point comparisons used in every input set $S_{i,d,n}$ are recorded as $CVF_{i,d,n}$ and $CMTF_{i,d,n}$ for two algorithms. Let $CVF_{d,n} = \frac{\sum_{i=1}^{10}(CVF_{i,d,n}/n)}{10}$ denote the average number of point comparisons used for VF algorithm and, correspondingly, let $CMTF_{d,n} = \frac{\sum_{i=1}^{10}(CMTF_{i,d,n}/n)}{10}$ for MTF algorithm. Fig. 2 shows the experimental results by $CVF_{d,n}$ and $CMTF_{d,n}$. It can be concluded from the figure that the VF algorithm has a better running performance than the MTF algorithm, if the point number n is small. And the average number of point comparisons of both algorithms tends to be the same constant as n increases. Bentley et al. [4] conjectured that the MTF algorithm runs in $N + o(N)$ expected time. This inspired us to give the following conjecture:

Conjecture 1. *The VF algorithm finds the maxima of N points chosen from a d-dimensional CI distribution in $O(N)$ time and it uses $N + o(N)$ point comparisons for any fixed dimensionality d.*

We prove this conjecture in the following section. It can be seen from the proof that the VF algorithm has a better theoretical result than the MTF algorithm does.



(a) 2-dimensional space.

(b) 3-dimensional space.

(c) 4-dimensional space.

(d) 5-dimensional space.

Fig. 2. Comparisons per point in 2, 3, 4 and 5-dimensional spaces.

4. Theoretical proof of the expected running time complexity

We introduce a method similar to the one presented in the literature [5] to prove Conjecture 1, where the idea is to adopt the probabilistic and amortized techniques. More specifically we prove the following:

Theorem 1. *Choose N points, $p_1, p_2, \ldots, p_N$, from a d-dimensional CI distribution. Then, with probability $1 - N^{-\Omega(\log N)}$, the VF algorithm finds their maxima in only $N + O(N^{2/3}\log^4 N)$ point comparisons when fixed $d = 2$; and $N + O(N^{d/(d+1)}\log^{d+1} N)$ point comparisons when $d > 2$.*

The $N^{-\Omega(\log N)}$ term can be thought of as being a super-polynomially small probability since it is smaller than N^{-k} for any constant k. It is the same probability as in Golin's proof of the Bentley's conjecture given in [5]. Like the discussion made in [5], we assume that each coordinate of the input points is uniformly distributed in [0, 1]. Under the same probability and input point condition, the VF algorithm uses only $N + O(N^{2/3}\log^4 N)$ point comparisons which is better than the MTF algorithm which uses $N + O(N^{6/7}\log^4 N)$ point comparisons. Note that there is no proved result for the MTF algorithm when the dimension is greater than 2. When the points are chosen from any CI distribution, it is not difficult to prove the same result by modifying the techniques through a mapping from CI distribution to [0, 1] uniform distribution. The approach to construct such a mapping is described as follows:

Suppose that $p_1, p_2, \ldots, p_N$ are chosen from some d-dimensional CI distribution. Let $F_i(x_i) = Pr(X \leq x_i)$ be the distribution function of the i-th component of points, where $i = 1, 2, \ldots, d$. For any given point $p = (x_1, x_2, \ldots, x_d)$, we define N, the natural mapping from the support of the CI distribution to the [0, 1] uniform distribution:

$$N(p) = N((x_1, x_2, \ldots, x_d)) = (F_1(x_1), F_2(x_2), \ldots, F_d(x_d)).$$

If $p_1, p_2, \ldots, p_N$ are chosen from any CI distribution, then $N(p_1), N(p_2), \ldots, N(p_N)$ have the same distribution as the points chosen from the [0, 1] uniform distribution. Because the volume of a point is based on the same mapping, the VF algorithm performs exactly the same sequence of operations on input $N(p_1), N(p_2), \ldots, N(p_N)$ as on input $p_1, p_2, \ldots, p_N$.

Our main result is Theorem 1. To give a formal proof of its validity we need to make a careful preparation, starting from the 2-dimensional case.

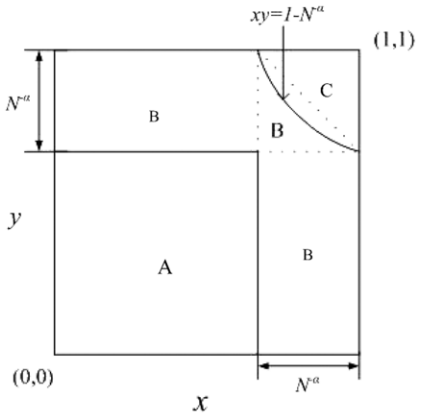


Fig. 3. The partition of the area of input points.

4.1. Proof of the theorem in 2-dimensional space

4.1.1. Random regions and variables

Let $p_1, p_2, \ldots, p_N$ be the input points listed in the order in which they are examined in 2-dimensional space. We partition the area of input points into three regions, A, B, C, dependent upon the parameter α as shown in Fig. 3, where $\alpha > 0$.

Formally

$$A = [0, 1 - N^{-\alpha}] \times [0, 1 - N^{-\alpha}],$$
$$C = \{(x, y) : xy \geq 1 - N^{-\alpha}, 1 - N^{-\alpha} \leq x \leq 1, 1 - N^{-\alpha} \leq y \leq 1\},$$
$$B = [0, 1] \times [0, 1] - A - C.$$

We also need some random variables which are functions of the input points and the regions. Set

$$F_C = \min_{1 \leq i \leq N} \{i : p_i \in C \text{ or } i = N\},$$
$$N_{B\,C} = |\{\, p_i : p_i \in B \cup C \,\}|,$$
$$M_i = |\{j : j \leq i \leq N, p_j \text{ is maximal in } p_1, \ldots, p_i\}|,$$
$$M = \max_{i \leq N} M_i.$$

In the above F_C is the index of the first point in C (if there is no such point, then $F_C = N$); $N_{B\,C}$ is the number of points found in the region $B \cup C$; M_i is the number of maxima in the point set $\{p_1, \ldots, p_i\}$; M is the largest of the M_i, it is an upper bound on the number of point comparisons that can be performed while examining any point to see if it is a maximal point.

4.1.2. Proofs of Lemmas 1–3

Lemma 1. *The number of point comparisons performed by VF algorithm, when running on a sequence of N d-dimensional points, $p_1, p_2, \ldots, p_N$, is at most $N + M \cdot F_c + M \cdot N_{B\,C}$.*

Proof. We partition the input sequence $p_1, p_2, \ldots, p_N$ into two subsequences $\{p_1, p_2, \ldots, p_{F_C}\}$ and $\{p_{F_C+1}, p_{F_C+2}, \ldots, p_N\}$.

We will show that the number of point comparisons performed while examining each subsequence can be expressed by functions of random variables F_C, $N_{B\,C}$ and M. These random variables are defined in Sections 4.1.1, 4.2.1 and 4.3.1 based on their dimensions $d = 2$, $d = 3$ and $d > 2$ respectively.

Claim 1. The total number of point comparisons needed to examine $p_1, p_2, \ldots, p_{F_C}$ is at most $M \cdot F_C$.

Claim 2. The total number of point comparisons needed to examine $p_{F_C+1}, p_{F_C+2}, \ldots, p_N$ is at most $N + M \cdot N_{B\,C}$.

Since the number of points in subsequence $p_1, p_2, \ldots, p_{F_C}$ is F_C, the total number of point comparisons needed to examine this subsequence is at most $M \cdot F_C$. After examining this subsequence the point p_{F_C} will be inserted into T and be located in the frontmost position of T, because the volume of point P_{F_C} is the largest one currently. After that, the frontmost position of T can only be replaced by other points in C, since the points in T are listed according to the descending order of points' volume and only the points in C can have larger volumes. Thus the frontmost point of T will always be in C. When examining $p_i \in \{p_{F_C+1}, p_{F_C+2}, \ldots, p_N\}$, there are only two cases: first, if $p_i \in A$, the number of point comparisons needed to examine p_i is 1, because the frontmost point of T is in C and it can dominate p_i. The number of such p_i is at most N; second, if $p_i \in B \cup C$, the number of point comparisons needed to examine p_i is at most M. The number of such p_i is $N_{B\,C}$. So the total number of point comparisons needed to examine $p_{F_C+1}, p_{F_C+2}, \ldots, p_N$ is at most $N + M \cdot N_{B\,C}$. The above discussion implies that Claims 1 and 2 are correct. Combining Claims 1 and 2, Lemma 1 is proved. □

Lemma 2. *Let $X_1, X_2, \ldots, X_n$ be $0-1$ random variables, and let $Pr(X_i = 1) = pr_i$, $X = \sum_{i=1}^{n} X_i$. Denote*

$$\mu = E(X) = E\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} E(X_i) = \sum_{i=1}^{n} pr_i.$$

If $E(\prod_{i=1}^{n} X_i) \leq \prod_{i=1}^{n} E(X_i)$, then for any $\varepsilon \geq e^2$, $Pr(X \geq \varepsilon\mu) < e^{-\varepsilon\mu}$.

Proof. According to the Markov's inequality, for any $t > 0$, we have

$$Pr(X \geq \varepsilon\mu) = Pr(e^{tX} \geq e^{t\varepsilon\mu}) \leq \frac{E(e^{tX})}{e^{t\varepsilon\mu}}$$
$$\leq \frac{\prod_{i=1}^{n} E(e^{tX_i})}{e^{t\varepsilon\mu}} = \frac{\prod_{i=1}^{n} (pr_i e^t + 1 - pr_i)}{e^{t\varepsilon\mu}} = \frac{\prod_{i=1}^{n} (1 + pr_i(e^t - 1))}{e^{t\varepsilon\mu}}.$$

For any y, $1 + y \leq e^y$, the above inequality

$$\leq \frac{\prod_{i=1}^{n} e^{pr_i(e^t-1)}}{e^{t\varepsilon\mu}} = \frac{\exp\left\{\sum_{i=1}^{n} pr_i(e^t - 1)\right\}}{e^{t\varepsilon\mu}} = \frac{e^{\mu(e^t-1)}}{e^{t\varepsilon\mu}}$$
$$= e^{\mu(e^t-1)-t\varepsilon\mu} = (e^{e^t-1-t\varepsilon})^{\mu} = (e^{t\varepsilon-e^t+1})^{-\mu} = (e^{t-e^t/\varepsilon+1/\varepsilon})^{-\varepsilon\mu}.$$

Let $t = log(\varepsilon) > 0$, and because $\varepsilon \geq e^2$, the above inequality

$$= (e^{log(\varepsilon)-1+1/\varepsilon})^{-\varepsilon\mu} \leq (e^{2-1+1/\varepsilon})^{-\varepsilon\mu} = (e^{1+1/\varepsilon})^{-\varepsilon\mu} < e^{-\varepsilon\mu}. \quad \square$$

Lemma 3. *Let $\alpha > 0$, $p_1, p_2, \ldots, p_N$ be a sequence of N 2-dimensional points chosen from the uniform distribution over the unit square. Then*

$$Pr(F_C > 2N^{2\alpha} \log^2 N) < N^{-\Omega(\log N)},$$
$$Pr(N_{B\,C} > 2e^2 N^{1-\alpha}) < N^{-\Omega(\log N)},$$
$$Pr(M > \log^2 N) < N^{-\Omega(\log N)}.$$

Proof. We first prove

$$Pr(F_C > 2N^{2\alpha} \log^2 N) < N^{-\Omega(\log N)}.$$

Suppose that q is a point chosen from the uniform distribution over the unit square. Then we see that

$$Pr(q \in C) = Area(C) = \int_0^{N^{-\alpha}} \int_0^{1-(1-N^{-\alpha})/(1-x)} dydx > \int_0^{N^{-\alpha}} \int_0^{N^{-\alpha}-x} dydx = \frac{1}{2}N^{-2\alpha}.$$

Then

$$Pr(F_C > j) = Pr(p_i \notin C, 1 \leq i \leq j) = (1 - Pr(q \in C))^j < \left(1 - \frac{1}{2}N^{-2\alpha}\right)^j,$$

so

$$Pr(F_C > 2N^{2\alpha} \log^2 N) < \left(1 - \frac{1}{2}N^{-2\alpha}\right)^{2N^{2\alpha} \log^2 N} = N^{-\Omega(\log N)}.$$

Continuing the above proof, the probability $q \in B \cup C$ is

$$p = Area(B \cup C) = 1 - (1 - N^{-\alpha})^2 = 2N^{-\alpha} - N^{-2\alpha}.$$

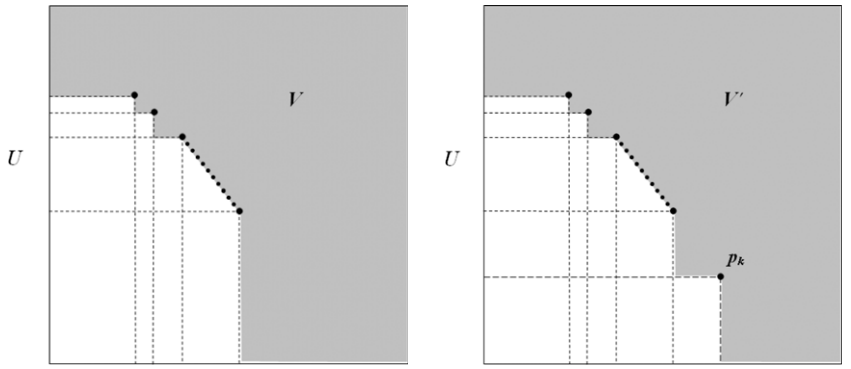
$N_{B\,C}$ is a binomially distributed random variable with parameters N and p, and

$$Pr(N_{B\,C} = i) = \binom{N}{i} p^i (1-p)^{N-i},$$

the expectation of $N_{B\,C}$ is $2N^{1-\alpha} - N^{1-2\alpha}$.

By Lemma 2, setting $\varepsilon = e^2$ implies that

$$Pr(N_{B\,C} > 2e^2 N^{1-\alpha}) = Pr(N_{B\,C} > e^2(2N^{1-\alpha} - N^{1-2\alpha}) + e^2 N^{1-2\alpha})$$
$$< Pr(N_{B\,C} > e^2(2N^{1-\alpha} - N^{1-2\alpha})) < \exp\{-e^2(2N^{1-\alpha} - N^{1-2\alpha})\} = N^{-\Omega(\log N)}.$$



(a) If p_k is not a maximal point. (b) If p_k is a maximal point.

Fig. 4. An example of the point set in 2-dimensional space. Black points denote currently maximal points.

The second inequality is thus proved. Now we turn to the last one: we first show that Lemma 2 can be applied in our discussion. Let $p_1, p_2, \ldots, p_N$ be the input points listed in the order in which they are examined. For a fixed $i \leq N$, let

$$X_j = \begin{cases} 1 & p_j \text{ is a maximal point in } p_1, p_2, \ldots, p_i\,; \\ 0 & \text{Otherwise} \end{cases}$$

be $0-1$ random variables. Then $\sum_{j=1}^{i} X_j$ is the number of maximal points in the point set $p_1, p_2, \ldots, p_i$, i.e., $M_i = \sum_{j=1}^{i} X_j$. For any $1 < (k+1=l) \leq i$,

$$\begin{aligned} E(X_kX_l) - E(X_k)E(X_l) &= Pr(X_k = 1 \cap X_l = 1) - Pr(X_k = 1)Pr(X_l = 1) \\ &= Pr(X_k = 1)Pr(X_l = 1|X_k = 1) - Pr(X_k = 1)[Pr(X_k = 1)Pr(X_l = 1|X_k = 1) + Pr(X_k = 0)Pr(X_l = 1|X_k = 0)] \\ &= Pr(X_k = 1)[Pr(X_l = 1|X_k = 1) - Pr(X_k = 1)Pr(X_l = 1|X_k = 1) - Pr(X_k = 0)Pr(X_l = 1|X_k = 0)] \\ &= Pr(X_k = 1)[(1 - Pr(X_k = 1))Pr(X_l = 1|X_k = 1) - Pr(X_k = 0)Pr(X_l = 1|X_k = 0)] \\ &= Pr(X_k = 1)Pr(X_k = 0)[Pr(X_l = 1|X_k = 1) - Pr(X_l = 1|X_k = 0)]. \end{aligned}$$

Since $k+1 = l$, it is seen that the VF algorithm examines the point p_k and then examines the point p_l. If p_k is a maximal point then the probability of p_l is a maximal point is less than the case that p_k is not a maximal point. An example of the point set in 2-dimensional space is seen from Fig. 4, where the black points denote the current maximal points, U denotes the area of input points, V and V' denote the shadow regions in Fig. 4(a) and (b), respectively, and $Area(V') \leq Area(V)$. It can be seen from the figure that

$$Pr(X_l = 1|X_k = 0) = Pr(p_l \text{ is a maximal point } | p_k \text{ is not a maximal point}) = \frac{Area(V)}{Area(U)},$$

$$Pr(X_l = 1|X_k = 1) = Pr(p_l \text{ is a maximal point } | p_k \text{ is a maximal point}) = \frac{Area(V')}{Area(U)} \leq \frac{Area(V)}{Area(U)}.$$

So $Pr(X_l = 1|X_k = 1) - Pr(X_l = 1|X_k = 0) \leq 0$, $E(X_kX_l) - E(X_k)E(X_l) \leq 0$. Then $E(\prod_{j=1}^{i} X_j) \leq \prod_{j=1}^{i} E(X_j)$. This explain that we can use Lemma 2 to prove the final inequality of Lemma 3.

It is known that the expectation of M_i is $E(M_i) = \Theta(\log^{d-1} i)$ for $i \leq N$ points chosen from any d-dimensional CI distribution [9]. We do not know what constant implicit in the $\Theta()$ notation is, but we can set this constant to δ, then $\mu = E(M_i) = \delta \log^{d-1} i$. Setting $\varepsilon = \frac{1}{\delta \log i} \log^2 N$ for dimension $d = 2$, if N is large enough, ε will $\geq e^2$. Substituting μ and ε into Lemma 2, we get

$$Pr(M_i > \log^2 N) < e^{-\log^2 N} = N^{-\Omega(\log N)}.$$

Since $M = \max_{i \leq N} M_i$,

$$Pr(M > \log^2 N) < e^{-\log^2 N} = N^{-\Omega(\log N)}. \quad \square$$

4.1.3. Proof of Theorem 1 when $d = 2$ using Lemmas 1 and 3

Proof. In Lemmas 1 and 3, the constants implicit in the $\Omega()$ notation are dependent only upon α. These two lemmas separate the deterministic part of the analysis from the probabilistic part. Inserting the probabilistic bounds of Lemma 3 into the deterministic one of Lemma 1 yields that the VF algorithm performs

$$N + O(\max(N^{2\alpha} \log^4 N, N^{1-\alpha} \log^2 N)) \tag{1}$$

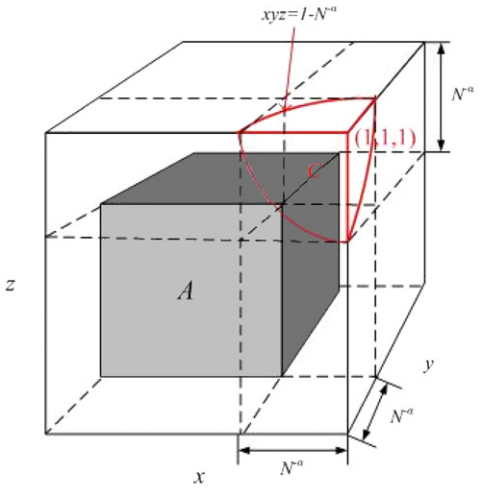


Fig. 5. Area partitions of input points in 3-dimensional space.

point comparisons with probability $1 - N^{-\Omega(\log N)}$. To get a simpler bound we choose $\alpha = 1/3$. Then the expression (1) becomes $N + O(N^{2/3} \log^4 N)$. So with probability $1 - N^{-\Omega(\log N)}$, the VF algorithm finds their maxima of N 2-dimensional points using only $N + O(N^{2/3} \log^4 N)$ point comparisons and so Theorem 1 is proved when the dimension $d = 2$. □

In the next subsection, we will use the same methods to prove Theorem 1 when the dimension $d = 3$.

4.2. Proof of Theorem 1 in 3-dimensional space

4.2.1. Random regions and variables

Let $p_1, p_2, \ldots, p_N$ be the input points listed in the order in which they are examined in 3-dimensional space. We also partition the area of input points into three regions, A, B, C, dependent upon the parameter α as shown in Fig. 5. The area of input points is $[0, 1] \times [0, 1] \times [0, 1]$. Formally

$$A = [0, 1 - N^{-\alpha}] \times [0, 1 - N^{-\alpha}] \times [0, 1 - N^{-\alpha}],$$
$$C = \{(x, y, z) : xyz \geq 1 - N^{-\alpha}, 1 - N^{-\alpha} \leq x \leq 1, 1 - N^{-\alpha} \leq y \leq 1, 1 - N^{-\alpha} \leq z \leq 1\},$$
$$B = [0, 1] \times [0, 1] \times [0, 1] - A - C.$$

In the above the random variables F_C, $N_{B\,C}$, M_i, M are defined in the same manner as in Section 4.1.1.

4.2.2. Proof of Lemma 4

Lemma 4. *Let $\alpha > 0$, $p_1, p_2, \ldots, p_N$ be a sequence of N 3-dimensional points chosen from the uniform distribution over the unit cube. Then*

$$Pr(F_C > 6N^{3\alpha} \log^2 N) < N^{-\Omega(\log N)},$$
$$Pr(N_{B\,C} > 3e^2 N^{1-\alpha}) < N^{-\Omega(\log N)},$$
$$Pr(M > e^2 \log^2 N) < N^{-\Omega(\log N)}.$$

Proof. First we prove

$$Pr(F_C > 6N^{3\alpha} \log^2 N) < N^{-\Omega(\log N)}.$$

Suppose that q is a point chosen from the uniform distribution over the unit cube. To bound $F_C = \min_{1 \leq i \leq N}\{i : p_i \in C \text{ or } i = N\}$ we set

$$p = Pr(q \in C) = Area(C) = \int_0^{N^{-\alpha}} \int_0^{1-(1-N^{-\alpha})/(1-x)} \int_0^{1-(1-N^{-\alpha})/((1-x)(1-y))} dzdydx$$
$$> \int_0^{N^{-\alpha}} \int_0^{N^{-\alpha}-x} \int_0^{N^{-\alpha}-x-y} dzdydx = \frac{1}{6} N^{-3\alpha}.$$

Then

$$Pr(F_C > j) = Pr(p_i \notin C, 1 \leq i \leq j) = (1 - p)^j < \left(1 - \frac{1}{6} N^{-3\alpha}\right)^j,$$

so

$$Pr(F_C > 6N^{3\alpha}\log^2 N) < \left(1 - \frac{1}{6}N^{-3\alpha}\right)^{6N^{3\alpha}\log^2 N} = N^{-\Omega(\log N)}.$$

Second, we prove

$$Pr(N_{B\,C} > 3e^2N^{1-\alpha}) < N^{-\Omega(\log N)}.$$

The expectation of $N_{B\,C}$ is $3N^{1-\alpha} - 3N^{1-2\alpha} + N^{1-3\alpha}$. Setting $\varepsilon = e^2$ in Lemma 2, then

$$\begin{aligned} Pr(N_{B\,C} > 3e^2N^{1-\alpha}) &= Pr(N_{B\,C}) > e^2(3N^{1-\alpha} - 3N^{1-2\alpha} + N^{1-3\alpha}) + e^2(3N^{1-2\alpha} - N^{1-3\alpha}) \\ &< Pr(N_{B\,C} > e^2(3N^{1-\alpha} - 3N^{1-2\alpha} + N^{1-3\alpha})) < \exp\{-e^2(3N^{1-\alpha} - 3N^{1-2\alpha} + N^{1-3\alpha})\} = N^{-\Omega(\log N)}. \end{aligned}$$

Next we prove the final inequality given in Lemma 4,

$$Pr(M > e^2\log^2 N) < N^{-\Omega(\log N)}.$$

The expectation M_i is $E(M_i) = \Theta(\log^2 i) = \delta\log^2 i$ for $i \le N$ points chosen from any 3-dimensional CI distribution. Setting $\mu = E(M_i) = \delta\log^2 i$ and $\varepsilon = e^2\frac{1}{\delta}\frac{\log^2 N}{\log^2 i}$, if N is large enough, ε will $\ge e^2$. Substituting μ and ε into Lemma 2, we get

$$Pr(M_i > e^2\log^2 N) < e^{-e^2\log^2 N} = N^{-\Omega(\log N)}.$$

Since $M = \max_{i\le N} M_i$,

$$Pr(M > e^2\log^2 N) < e^{-e^2\log^2 N} = N^{-\Omega(\log N)}. \quad \square$$

4.2.3. Proof of Theorem 1 when $d = 3$ using Lemmas 1 and 4

Proof. The proof is similar to the 2-dimensional case. Lemmas 1 and 4 separate the deterministic part of the analysis from the probabilistic part. Inserting the probabilistic bounds of Lemma 4 into the deterministic one of Lemma 1 yields that the VF algorithm performs

$$N + O(\max(N^{3\alpha}\log^4 N, N^{1-\alpha}\log^2 N)) \tag{2}$$

point comparisons with probability $1 - N^{-\Omega(\log N)}$. To get a simpler bound we choose $\alpha = 1/4$. Then the expression (2) becomes $N + O(N^{3/4}\log^4 N)$. So with probability $1 - N^{-\Omega(\log N)}$, the VF algorithm finds their maxima of N 3-dimensional points using only $N + O(N^{3/4}\log^4 N)$ point comparisons. $\square$

We now have proven Theorem 1 when the dimensions are $d = 2$ and $d = 3$, and found a general method to prove those inequations. In the following subsection, we will prove Theorem 1 for any dimension d (>2) to finish the theoretical proof of the expected running time.

4.3. Proof of Theorem 1 in d-dimensional Space ($d > 2$)

4.3.1. Random regions and variables

Let $p_1, p_2, \ldots, p_N$ be the input points listed in the order in which they are examined in d-dimensional space and $d > 2$. We again partition the area of input points into three regions, A, B, C, dependent upon the value of parameter α. The area of input points is $[0, 1]^d$. Let the coordinates be $x_1, x_2, \ldots, x_d$. Formally

$$\begin{aligned} &A = [0, 1 - N^{-\alpha}]^d, \\ &C = \{(x_1, x_2, \ldots, x_d) : x_1x_2\cdots x_d \ge 1 - N^{-\alpha}, 1 - N^{-\alpha} \le x_1 \le 1, 1 - N^{-\alpha} \le x_2 \le 1, \ldots, 1 - N^{-\alpha} \le x_d \le 1\}, \\ &B = [0, 1]^d - A - C. \end{aligned}$$

In the above the random variables F_C, $N_{B\,C}$, M_i, M are defined in the same manner as in Section 4.1.1.

4.3.2. Proof of Lemma 5

Lemma 5. *Let $\alpha > 0$, $d > 2$, $p_1, p_2, \ldots, p_N$ be a sequence of N d-dimensional points chosen from the uniform distribution over the unit hypercube. Then*

$$\begin{aligned} &Pr(F_C > d!N^{d\alpha}\log^2 N) < N^{-\Omega(\log N)}, \\ &Pr(N_{B\,C} > e^2dN^{1-\alpha}) < N^{-\Omega(\log N)}, \\ &Pr(M > e^2\log^{d-1} N) < N^{-\Omega(\log N)}. \end{aligned}$$

Proof. First we prove

$$Pr(F_C > d!N^{d\alpha}\log^2 N) < N^{-\Omega(\log N)}.$$

Suppose that q is a point chosen from the uniform distribution over the unit hypercube. To bound $F_C = \min_{1\le i\le N}\{i : p_i \in C \text{ or } i = N\}$ we set

$$\begin{aligned} p &= Pr(q \in C) = Area(C) \\ &= \int_0^{N^{-\alpha}} \int_0^{1-(1-N^{-\alpha})/(1-x_1)} \int_0^{1-(1-N^{-\alpha})/((1-x_1)(1-x_2))} \cdots \int_0^{1-(1-N^{-\alpha})/((1-x_1)(1-x_2)\cdots(1-x_d))} dx_d \cdots dx_2 dx_1 \\ &> \int_0^{N^{-\alpha}} \int_0^{N^{-\alpha}-x_1} \int_0^{N^{-\alpha}-x_1-x_2} \cdots \int_0^{N^{-\alpha}-x_1-x_2\cdots-x_d} dx_d \cdots dx_2 dx_1 \\ &= \frac{1}{d!} N^{-d\alpha}. \end{aligned}$$

Then

$$Pr(F_C > j) = Pr(p_i \notin C, 1 \le i \le j) = (1-p)^j < \left(1 - \frac{1}{d!}N^{-d\alpha}\right)^j,$$

so

$$Pr(F_C > d!N^{d\alpha}\log^2 N) < \left(1 - \frac{1}{d!}N^{-d\alpha}\right)^{d!N^{d\alpha}\log^2 N} = N^{-\Omega(\log N)}.$$

Second, we prove

$$Pr(N_{B\,C} > e^2 dN^{1-\alpha}) < N^{-\Omega(\log N)}.$$

Expectation of $N_{B\,C}$ is $\sum_{i\ge 1}(-1)^{i-1}\binom{d}{i}N^{1-i\alpha}$, setting $\varepsilon = e^2$ in Lemma 2, then

$$\begin{aligned} Pr(N_{B\,C} > e^2 dN^{1-\alpha}) &= Pr\left(N_{B\,C} > e^2\left(dN^{1-\alpha} - \sum_{i\ge 2}(-1)^i\binom{d}{i}N^{1-i\alpha}\right) + e^2\sum_{i\ge 2}(-1)^i\binom{d}{i}N^{1-i\alpha}\right) \\ &< Pr\left(N_{B\,C} > e^2\left(dN^{1-\alpha} - \sum_{i\ge 2}(-1)^i\binom{d}{i}N^{1-i\alpha}\right)\right) \\ &= Pr\left(N_{B\,C} > e^2\sum_{i\ge 1}(-1)^{i-1}\binom{d}{i}N^{1-i\alpha}\right) < \exp\left\{-e^2\sum_{i\ge 1}(-1)^{i-1}\binom{d}{i}N^{1-i\alpha}\right\} = N^{-\Omega(\log N)}. \end{aligned}$$

Next we prove the final inequality given in Lemma 5,

$$Pr(M > e^2\log^{d-1} N) < N^{-\Omega(\log N)}.$$

The expectation M_i is $E(M_i) = \Theta(\log^{d-1} i) = \delta\log^{d-1} i$ for $i \le N$ points chosen from any d-dimensional CI distribution. Setting $\mu = E(M_i) = \delta\log^{d-1} i$ and $\varepsilon = e^2\frac{1}{\delta}\frac{\log^{d-1} N}{\log^{d-1} i}$, if N is large enough, ε will $\ge e^2$. Substituting μ and ε into Lemma 2, we get

$$Pr(M_i > e^2\log^{d-1} N) < e^{-e^2\log^{d-1} N} = N^{-\Omega(\log N)}.$$

Since $M = \max_{i\le N} M_i$,

$$Pr(M > e^2\log^{d-1} N) < e^{-e^2\log^{d-1} N} = N^{-\Omega(\log N)}. \quad \square$$

4.3.3. Proof of Theorem 1 when $d > 2$ using Lemmas 1 and 5

Proof. Inserting the probabilistic bounds of Lemma 5 into the deterministic one of Lemma 1 yields that the VF algorithm performs at most

$$N + e^2\log^{d-1} Nd!N^{d\alpha}\log^2 N + e^2\log^{d-1} Ne^2 dN^{1-\alpha} = N + e^2 d!N^{d\alpha}\log^{d+1} N + e^4 dN^{1-\alpha}\log^{d-1} N \tag{3}$$

point comparisons with probability $1 - N^{-\Omega(\log N)}$. For a fixed d, the expression (3) is equivalent to

$$N + O(\max(N^{d\alpha}\log^{d+1} N, N^{1-\alpha}\log^{d-1} N)). \tag{4}$$

To get a simpler bound we choose $\alpha = 1/(d+1)$, then the expression (4) becomes $N + O(N^{d/(d+1)}\log^{d+1} N)$. So with probability $1 - N^{-\Omega(\log N)}$, the VF algorithm can find the maxima of N d-dimensional points in only $N + O(N^{d/(d+1)}\log^{d+1} N)$ point comparisons and Theorem 1 is therefore proven for $d > 2$. $\square$

Combining all the above discussion a complete theoretical proof of Theorem 1 is thus done.

5. Conclusion

In this paper we presented a volume first maxima-finding algorithm. Experimental data and theoretical analysis show that it runs faster than the MTF algorithm, if the points are chosen from CI distribution. Under such distribution, we also proved that with probability $1 - N^{-\Omega(\log N)}$, the expected running time of this algorithm is only $N + O(N^{2/3} \log^4 N)$ in the 2-dimensional space, and $N + O(N^{d/(d+1)} \log^{d+1} N)$ in d (>2) dimensional space.

In the paper random inputs are chosen from CI distribution. How well does the VF algorithm work when the points are chosen from a non-CI distribution would be an interesting question. We tried to consider the problem, but we could not make it. We therefore leave it as an open question in this note.

Acknowledgement

The authors would like to thank the referees for their several insightful comments and invaluable suggestions toward improving the presentation of this paper.

References

[1] W.M. Chen, H.K. Hwang, T.H. Tsai, Efficient Maxima-finding algorithms for random planar samples, Discrete Mathematics and Theoretical Computer 6 (2003) 107–122.
[2] F.P. Preparata, M.I. Shamos, Computational Geometry: An Introduction, Springer-Verlag, New York, 1985.
[3] H.T. Kung, F. Luccio, F.P. Preparata, On finding the maxima of a set of vectors, Journal of the ACM 22 (1975) 469–476.
[4] J.L. Bentley, K.L. Clarkson, D.B. Levine, Fast linear expected-time algorithms for computing maxima and convex Hulls, in: SODA, 1990, pp. 179–187.
[5] M.J. Golin, A provably fast linear-expected-time maxima-finding algorithm, Algorithmica 11 (1994) 501–524.
[6] S. Börzsönyi, D. Kossmann, K. Stocker, The skyline operator, in: Proceedings of the ICDE, 2001, pp. 421–430.
[7] J. Chomicki, P. Godfrey, J. Gryz, D. Liang, Skyline with presorting, in: Proceedings of the ICDE, 2003, pp. 717–719.
[8] P. Godfrey, R. Shipley, J. Gryz, Maximal vector computation in large data sets, in: Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005, pp. 229–240.
[9] J.L. Bentley, H.T. Kung, M. Schkolnick, C.D. Thompson, On the average number of maxima in a set of vectors and applications, Journal of the Association for Computing Machinery 25 (1978) 536–543.